CLAIMS

1. (Currently Amended) A method for resolving a partial media topology, comprising:

receiving a partial media topology that includes a plurality of nodes including at least one media source node and at least one media sink node;

populating a working first-in-first-out (FIFO) queue with source nodes in the partial topology; and

iteratively, for each node in the working FIFO queue:

negotiating a media type for each output of the node with a downstream node in the partial topology[[;]],

instantiating one or more intermediate nodes when it is determined that an output of the node is incompatible with an input of the downstream node;

connecting the one or more intermediate nodes between the media source node and the media sink node[[;]], ~~and~~

adding the one or more intermediate nodes to the working FIFO queue ~~only if all input connections of the intermediate nodes are resolved~~, the one or more intermediate nodes being absent from the partial media topology, and

removing one or more of the added intermediate nodes from the working FIFO queue when a connection between a first intermediate node of the working FIFO queue and a first media node of the working FIFO queue is unresolved.

2. (Original) The method of claim 1, wherein the partial media topology is received from a remote process as a parameter in an interface call.

3. (Previously Presented) The method of claim 1, wherein the working FIFO queue comprises each node in the partial topology, and wherein an ordering of the nodes in the partial topology is maintained from the partial topology to the working FIFO queue.

4. (Previously Presented) The method of claim 1, wherein negotiating a media type comprises determining media types of an upstream node and an associated downstream node.

5. (Original) The method of claim 1, wherein instantiating one or more intermediate nodes comprises instantiating at least one of an encoder or a decoder.

6. (Previously Presented) The method of claim 1, wherein adding the one or more intermediate nodes to the working FIFO queue comprises adding one or more intermediate nodes to convert a compressed output stream of the source node into an uncompressed output.

7. (Original) The method of claim 5, wherein the encoder converts an uncompressed media stream into a compressed media stream.

8. (Original) The method of claim 1, wherein connecting the one or more intermediate nodes between the upstream node and the downstream node comprises generating a data path between the output of a upstream node an input of an intermediate node.

9. (Original) The method of claim 1, wherein one or more of the intermediate nodes is an option node.

10. (Currently Amended) A system comprising:

one or more tangible computer-readable media;

a media engine embodied on the one or more computer-readable media and configured to communicatively interact with an application to present a media presentation;

the media engine being configured to use:

a media session to generate a partial topology, the partial topology including one or more media sources individual ones of which serving as a source of media content, and one or more media sinks configured to sink a media stream, and

a topology loader to resolve the partial topology into a full topology, wherein the topology loader resolves the partial topology, in part, by: (1) inserting one or more intermediate nodes into a first-in-first-out (FIFO) queue, and (2) removing one or more inserted intermediate nodes from the FIFO queue when a connection between a first inserted intermediate node of the

FIFO queue and a first media node of the FIFO queue is unresolved~~a count of nodes in the full topology is greater than a count of nodes in the partial topology~~.

11. (Original) The system of claim 10, wherein the media engine exposes one or more application program interfaces that are used by an application to interact directly with the media engine, and indirectly with components used by the media engine.

12. (Original) The system of claim 10, wherein the media session invokes the topology loader using an application programming interface.

13. (Original) The system of claim 10, wherein the media session passes the partial topology to the topology loader as a parameter in an interface call.

14. (Currently Amended) The system of claim 10, wherein the topology loader is configured to ~~instantiate one or more intermediate nodes, and to~~ connect at least one of the one or more intermediate nodes in a communication path between a media source and a media sink in the partial topology.

15. (Original) The system of claim 14, wherein the one or more intermediate nodes comprise a decoder for decoding the output of a source node.

16. (Previously Presented) The system of claim 14, wherein the one or more intermediate nodes comprise an encoder for encoding an input of a source node.

17. (Original) The system of claim 14, wherein the one or more intermediate nodes comprise an optional node, and wherein the topology loader implements logic to connect an optional node.

18. (Original) The system of claim 10, wherein the topology loader provides at least one interface to provide the application the capability to facilitate resolving the partial topology.

19. (Currently Amended) The system of claim 10, wherein the topology loader returns [[a]]the fully ~~resolved~~ topology.

20. (Currently Amended) A system comprising:

one or more tangible computer-readable media;

a media engine embodied on the one or more computer-readable media and configured to communicatively interact with an application to present a presentation;

the media engine being configured to use:

a media session to generate one or more media sources individual ones of which serving as a source of media content, and one or more media sinks configured to sink a media stream;

a topology loader to generate one or more transforms communicatively linked with one or more media sources and configured to operate on data received from the one or more media sources, the topology loader to further receive a partially resolved topology from the media session, and to generate a fully resolved topology by sequentially negotiating a media type of each source node of the partially resolved topology with an input of a downstream node to determine whether additional intermediate nodes should be added, and to remove at least one added intermediate node when a connection between an added intermediate node and a source node is unresolved.

21. (Previously Presented) The system of claim 20, wherein

the media session creates the partial topology, the partial topology to present the presentation.

22. (Original) The system of claim 21, wherein the media engine creates partial topology by at least determining one or more media sources and one or more media sinks for the presentation.

23. (Original) The system of claim 20, wherein the topology loader analyzes the outputs of a media source and the inputs of a media sink, and negotiates the media type for passing a media stream between the media source and the media sink.

24. (Original) The system of claim 20, wherein the topology loader generates a source node list comprising nodes in the partial topology.

25. (Original) The system of claim 24, wherein the one or more transforms generated by the topology loader are added to the source node list.

26. (Original) The system of claim 25, wherein the topology loader negotiates the media type between the one or more transforms and one or more downstream nodes.

27. (Previously Presented) The system of claim 20, wherein the one or more transforms comprise at least of an encoder or a decoder.

28. (Original) The system of claim 20, wherein the topology loader returns the fully resolved topology to the media session.

29. (Previously Presented) A method for resolving a partial media topology, comprising:

receiving a plurality of media nodes from a remote computer;

populating a working first-in-first-out (FIFO) queue with at least two of the plurality of media nodes;

resolving a connection between two nodes of the FIFO queue by adding an encoder to the FIFO queue;

inserting one or more intermediate nodes into the FIFO queue, the one or more added intermediate nodes absent from the plurality of media received from the remote computer;

removing one or more of the inserted nodes from the FIFO queue when a connection between a first intermediate node of the FIFO queue and a first media node of the FIFO queue is unresolved;

generating a presentation media comprised of the FIFO nodes; and

presenting the presentation media to a user by visually rendering the FIFO nodes within a window on a display device.